Université Mohammed V de Rabat
Faculté des Sciences
SMP4 – Printemps 2015

T.D. N°3

Exercice 1

```
#include <stdio.h>

main()
{
   struct date
   {
      int jj;
      int mm;
      int aa;
   };
   typedef struct date Date;
   Date naissance;
   /* saisie sans controle ...*/
   printf("Donner votre date de naissance (j m a) ");
   scanf("%d",&naissance.jj);
   scanf("%d",&naissance.mm);
   scanf("%d",&naissance.aa);
   /* affichage ...*/
   printf("Vous etes ne le %d/%d/%d",naissance.jj,naissance.mm,naissance.aa);
}
```

Exercice 2

```
#include <stdio.h>

main()
{
   typedef struct { int jj,mm,aa;} Date;
   struct fiche
   {
      char nom[30];
      char prenom[30];
      char tel[14];
      char email[50];
      Date ddn;
      long cne;
   };
   typedef struct fiche Fiche;

   Fiche copains[100];
   int i;
   /* saisie sans controle ...*/
   for(i=0;i<100; i++)
   {
      printf("Information de la personne Num %d\n",i+1);
      printf("Donner le Code: ");
        scanf("%d",&copains[i].cne);
      printf("Donner le nom: ");
        scanf("%s",&copains[i].nom);
```

```
        printf("Donner le prenom: ");
          scanf("%s",&copains[i].prenom);
        printf("Donner le Num de Tel: ");
          scanf("%s",&copains[i].tel);
        printf("Donner l email: ");
          scanf("%s",&copains[i].email);
        printf("Donner la date de naissance: ");
          scanf("%d",&copains[i].ddn.jj);
          scanf("%d",&copains[i].ddn.mm);
          scanf("%d",&copains[i].ddn.aa);
    }
    /* affichage ...
    ...........
    ....*/
}
```

Exercice 3

	A	**B**	**C**	**P1**	**P2**
départ	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C	1	2	3	&A	&C
**P1=(*P2)++*	3	2	4	&A	&C
P1=P2	3	2	4	&C	&C
P2=&B	3	2	4	&C	&B
**P1-=*P2*	3	2	2	&C	&B
*++*P2*	3	3	2	&C	&B
**P1*=*P2*	3	3	6	&C	&B
*A=++*P2**P1*	24	4	6	&C	&B
P1=&A	24	4	6	&A	&B
**P2=*P1/=*P2*	6	6	6	&A	&B

Exercice 4

a) *P+2 => la valeur 14
b) *(P+2) => la valeur 34
c) &P+1 => l'adresse du pointeur derrière le pointeur P (rarement utilisée)
d) &A[4]-3 => l'adresse de la composante A[1]
e) A+3 => l'adresse de la composante A[3]
f) &A[7]-P => la valeur (indice) 7
g) P+(*P-10) => l'adresse de la composante A[2]
h) *(P+*(P+8)-A[7]) => la valeur 23

Exercice 5

```
#include <stdio.h>

main()
{
 double EXP1(double x, int n);
 void   EXP2(double *x, int n);

 double a;
 int b;

 printf("Donner un Nombre X : ");
 scanf("%lf", &a);
 do
 {
   printf("Donner l'exposant N (>=0) : ");
```

```
   scanf("%d", &b);
 } while(b<0);
 /* a */
 printf("EXP1(%.2f , %d) = %f\n", a, b, EXP1(a,b));
 /* b */
 printf("EXP2(%.2f , %d) = ", a, b);
 EXP2(&a, b);
 printf("%f\n", a);
}

double EXP1(double x, int n)
{
 double r;
 for (r=1.0; n>0; n--)
     r *= x;
 return r;
}

void EXP2(double *x, int n)
{
 double tmp;
 for (tmp=1.0; n>0; n--)
     tmp *= *x;
 *x = tmp;
}
```

Exercice 6

```
#include <stdio.h>
#define NMAX 50

void lireTabl (int *T, int *n, int x);

main()
{
    int A[NMAX],m,i;
    lireTabl(A,&m,NMAX);
    for(i=0;i<m;i++)
        printf("%d\n",A[i]);
}

void lireTabl (int *T, int *n, int x)
{
 int i;
 do
    {
     printf("Donner la taille du tableau (max.%d) : ", NMAX);
     scanf("%d", n); /* n et non &n ! */
    }while (*n<0 || *n>NMAX);
 for (i=0; i<*n; i++)
    {
     printf("Elément[%d] : ", i+1);
     scanf("%d", T+i);
    }
}
```

Exercice 7

```
#include <stdio.h>
```

```
#define LMAX 50
#define CMAX 50

void lireTabl2 (int *T, int *n, int N, int *m, int M);

main()
{
   int A[LMAX][CMAX],n,m,i,j;
   lireTabl2((int *)A,&n,LMAX,&m,CMAX);
/* ou lireTabl2(&A[0][0],&n,LMAX,&m,CMAX); */
   for(i=0;i<n;i++)
      for(j=0;j<m;j++)
      {
         printf("%d\n",A[i][j]);
         printf("\n");
      }
}

void lireTabl2 (int *T, int *n, int N, int *m, int M)
{
   int i,j;
 do
  {
   printf("Donner Nombre de lignes  (max.%d) : ",N);
   scanf("%d", n);
  }while (*n<0 || *n>N);
 do
  {
   printf("Donner Nombre de colonnes (max.%d) : ",M);
   scanf("%d", m);
  } while (*m<0 || *m>M);

 for (i=0; i<*n; i++)
     for (j=0; j<*m; j++)
        {
         printf("Elément(%d,%d) : ", i+1, j+1);
         scanf("%d", T + i*M + j);
        }
}
```

Exercice 8

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define NBL  30                        /*Nombre de Letres max dans un mots*/
#define N    10                     /* Nombre d'essais */

main()
{
  char *dico="dico.txt";               /* Nom du fichier */
  void enregistre_mots(char *);         /* ajouter des mots au dictionnaire*/
  int NbMotsDico(FILE *);               /* calcule le nombre de mots dans le Dictionnaire*/
  void jouer(char *);                  /* faire une partie de plusieurs mots... */
  int saisie_mot(char *chaine, int longueur);
```

```
  void viderBuffer();
  void cache_mot(char *chaine, int longueur);
  int verifie_lettre_dans_mot(char l,char *chaine1,char *chaine2,int longueur);

  int i;                    /* indicateur menu */

do
{
    do
        {
         printf("faite un choix et taper:\n\t0 pour quitter\n\t1 pour ajouter des mots au dictionnaire\n\t2 pour jouer\nVotre choix? : ");
            scanf("%d",&i);
        } while(i<0 || i>2);
        switch (i)
        {
            case 0 :
                printf(" Merci d'avoir jouer avec nous\n");
                break;
            case 1 :
                enregistre_mots(dico);
                break;
            case 2 :
                jouer(dico);
                break;
        }
}while(i);
}

void viderBuffer()
{
    int c = 0;
    while (c != '\n' && c != EOF)
    {
        c = getchar();
    }
}

int saisie_mot(char *chaine, int longueur)
{
    int lng;
    char *positionEntree = NULL;
    printf("Entrez un mot : ");
    viderBuffer();
    if (fgets(chaine, longueur, stdin) != NULL)  /* Pas d'erreur de saisie ?*/
    {
        positionEntree = strchr(chaine, '\n');
        if (positionEntree != NULL)
        {
            *positionEntree = '\0';
        }
        else
        {
            viderBuffer();
        }
```

```
        lng=0;
        while( chaine[lng])
           {
              chaine[lng]=toupper(chaine[lng]);
              lng++;
           }
        return lng;
    }
    else
    {
        viderBuffer();
        return 0;
    }
}

void cache_mot(char *chaine, int longueur)
{
    int i;
    for(i=0;i<longueur;i++)
        chaine[i]='*';
    chaine[i]='\0';
}

int verifie_lettre_dans_mot(char l,char *chaine1,char *chaine2,int longueur)
{
    int i,j=i=0;
    while ((i < longueur))   /* recherche du caractere dans le mot caché */
        {
            if(chaine1[i] == l)    /* si lettre trouvee on met a jour le masque */
                chaine2[i]=l;
            if(chaine2[i]=='*')
                    j++;
            i++;
        }
    return j;
}
void enregistre_mots(char * dict)
{
    FILE *fichier=fopen(dict,"a");
    int i;
    char mot[NBL];                    /* mot a trouver */
    if(!fichier)
    {
         printf("probleme d'ouverture de %s\n",dict);
         return;
    }
    printf("Vous allez ajouter des mots au dictionnaire du pendu\n");
    do
    {
        i=saisie_mot(mot, NBL);
        fprintf(fichier,"%s\n",mot);
        printf("voulez vous ajouter un autre mot\?\n\t 0 pour Non\n\t Autre pour Oui\nVotre choix? : ");
        scanf("%d",&i);
    }while(i);
    fclose(fichier);
```

```
 }
int NbMotsDico(FILE *fichier)
{
   int i=0;
   char c;
   do
      {
        c = fgetc(fichier);
        if(c == '\n')
            i++;
      }while(c!= EOF);
   return i;
}

 void jouer(char *dict)
 {
    int i,nbmots;
    int positionmot;
    FILE *fichier=fopen(dict,"r");
    char mot[NBL];                  /* mot a trouver */
    char masque[NBL];               /* masque pour affichage */
    char lettre ;                   /* lettre proposee */
    int nbcoups;                    /*  indice pour boucle  */
    int nb_lettres_manquants;       /* nb de lettres  à trouvées */

    if(!fichier)
      {
        printf("probleme d'ouverture de %s\n",dict);
        return;
      }
      printf("Vous allez jouer au pendu\n");
      srand(time(NULL));
      nbmots=NbMotsDico(fichier);
      do
         {
           rewind(fichier);                    /* on se replace au debut du fichier..*/
           positionmot=(rand() % nbmots) + 1;
           for(i=0;i<positionmot;i++)
              fgets(mot,NBL,fichier);
           mot[strlen(mot)-1]='\0';              /* enlever le caractere séparateur espace qui termine le mot...*/
           cache_mot(masque, strlen(mot));       /* initialisation du masque avec des etoiles */
           printf("%s", masque);                 /* affichage du nombre d'etoiles representant le mot a trouver */
           nb_lettres_manquants=strlen(mot);
           nbcoups = 0;
           while ((nbcoups < N) && (nb_lettres_manquants>0)) /* on autorise N essais de 0 a N-1 */
              {
                printf ("\ncaractere ? ");           /*  demande d'un caractere */
                scanf (" %c", &lettre);
                lettre= toupper(lettre);             /* mise en majuscule */
                nb_lettres_manquants=verifie_lettre_dans_mot(lettre,mot,masque,strlen(mot));
                printf("%s", masque);                /*  reaffichage du masque */
                nbcoups++;                           /* incremente le nombre d'essais */
           }
           if (nb_lettres_manquants)                 /* on est sorti parce que nb coups > N */
              printf("\nVOUS AVEZ PERDU\n");
```

```
                    else
                    printf("\nVOUS AVEZ GAGNE en %d coups\n", nbcoups);
                printf("voulez vous rejouer avec un autre mot\?\n\t 0 pour Non\n\t autre pour Oui\n\nVotre choix?
: ");
                scanf("%d",&i);
            }while(i);
        fclose(fichier);
}
```

Exercice 9

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

main()
{
    char *chaine(void);

    char  *c=NULL;

    printf("Donner une chaine de caracteres: ");
    c=chaine();
    printf("Chaine retournee: %s",c);
    free(c);
    c=NULL;

}

char *chaine(void)
{
    char tmp[1000], *a=NULL;
    gets(tmp);
    a=(char *)malloc((strlen(tmp)+1)*sizeof(char));
    strcpy(a,tmp);
    return(a);
}
```